A hierarchical fuzzy cluster ensemble approach and its application to big data clustering

Su, Pan; Shang, Changjing; Shen, Qiang

Published in:
Journal of Intelligent and Fuzzy Systems

DOI:
10.3233/IFS-141518

Publication date:
2015

Citation for published version (APA):
Su, P., Shang, C., & Shen, Q. (2015). A hierarchical fuzzy cluster ensemble approach and its application to big data clustering. *Journal of Intelligent and Fuzzy Systems*, *28*(6), 2409-2421. https://doi.org/10.3233/IFS-141518

Undefined 0 (0) 1–3
IOS Press

A Hierarchical Fuzzy Cluster Ensemble Approach and its Application to Big Data Clustering

Pan Su [a], Changjing Shang [a,*] and Qiang Shen [a]
[a] *Department of Computer Science, Institute of Mathematics, Physics and Computer Science, Aberystwyth University, Aberystwyth, Ceredigion, SY23 3DB, UK*
E-mail: {pas23,cns,qqs}@aber.ac.uk

Abstract. Cluster ensembles organically integrate individual component methods which may utilise different parameter settings and features, and which may themselves be generated on the basis of different representations and learning mechanisms. Such a technique offers an effective means for aggregating multiple clustering results in order to improve the overall clustering accuracy and robustness. Many topics regarding cluster ensembles have been proposed and promising results are gained in the literature. To reinforce such development, this paper presents another cluster ensemble approach for fuzzy clustering, with an aim to be applied for clustering of big data. The proposed algorithm first generates fuzzy base clusters with respect to each data feature and then, employs a fuzzy hierarchical graph to represent the relationships between the resulting base clusters. Whilst the work employs fuzzy c-means and hierarchical clustering in generating base cluster and implementing consensus function respectively, when applied to large datasets it has lower time complexity than the original fuzzy c-means and hierarchical clustering. The resultant ensemble clustering mechanism is tested against traditional clustering methods on various benchmark datasets. Experimental results demonstrate that it generally outperforms crisp cluster ensembles and single linkage agglomerative clustering, in terms of accuracy in conjunction with time efficiency, thereby showing that it has the potential for application in clustering big data.

Keywords: Fuzzy cluster ensemble, big data clustering, fuzzy c-means, hierarchical clustering, data mining

1. Introduction

Dealing with big data has become inevitable in many real-world problems. Recently, a new trend of and indeed challenge for data mining has arisen with the exponential growth and also availability of large amount of complex data. Applying conventional data mining techniques directly to big data is difficult or even impossible due to its intolerable computational time. Besides, the high dimensional and multi-model features may degrade the performance of conventional learning algorithms [35]. A number of research directions have been proposed in the literature to overcome such difficulties, including re-sampling data and distributing or parallelising conventional algorithms [1,23].

Clustering is one of the important approaches within the framework of unsupervised learning which is helpful for finding hidden structures in unlabelled datasets. In general, the task of clustering is to assign objects to groups (namely clusters) such that data points in the same group are similar to each other, and dissimilar to those in the other clusters [15]. A good number of clustering algorithms have been proposed in the literature, and successfully applied to a range of problems [37,39]. However, clustering big data is more challenging than dealing with traditional data modelling and analysis problems. Many of the existing clustering methods such as the k-means and fuzzy c-means are NP-hard, and hence they are very time consuming for handling big data.

*Corresponding author. E-mail: cns@aber.ac.uk

To tackle the aforementioned problem feasible techniques have been proposed. Most of which work by extending the existing approaches (that have been developed for non big data) through analysing a selected, manageable amount of samples of the original data and then, exploiting the sample-based modelling results to derive a partition for the overall data. These methods differ usually only in terms of how the sample-based analysis is carried out, including the CLARA algorithm [19] and the CLARANS algorithm [24]. Clustering big data has also led to distributed and parallel implementations [26]. One such approach is to directly extend existing clustering methods by taking advantage of distributed network environments in which the overall computation effort is shared by collaborative computing facilities [45]. Whilst these promising results have been reported, much remains to be done in order to have a more efficient and effective fuzzy clustering approach that is suitable for big data. Inspired by this observation, a hierarchical fuzzy cluster ensemble (HFCE) method (which is applicable for distributed and parallel application) is proposed in this paper.

Cluster ensembles are a type of ensemble-based unsupervised learning technique which combine multiple individual clustering results into a single consolidated partition. They have been proposed to achieve accuracy and robustness superior to those of the individual clustering methods [31]. However, the performance of cluster ensembles generally depends on both the quality and the diversity of ensemble components [12,21].

Two essential steps are particularly identified that are commonly involved in the development of cluster ensembles: 1) the generation of ensemble components (i.e., base clustering methods), and 2) the consensus of them. To ensure diversity of component clustering results typical ensemble generation strategies include: different parameter configurations of a given clustering algorithm (such as different numbers of clusters and different cluster centre initialisations) [10], and random feature-space or sampling techniques [6,7,40]. Regarding the issue of consensus, most of the existing methods employ a form of instance-cluster matrix that summarises the results of base ensembles to achieve the final partition [25,34]. An interesting direction to further strengthen the performance of cluster ensembles is to embed the information contained within base clusters into a graph or a hypergraph such that consensus can be applied in a link-based manner [9,31]. Following this idea, recently, the link-based refinement methods have been proposed [14], where the links between base clusters are employed to refine the results of component clusters.

Apart from conventional clustering algorithms whose outputs are hard partitions of data, there are alternative approaches such as EM [5] and fuzzy c-means [4], which generate soft or fuzzy partitions of data with a natural appeal. In order to take advantage of the aforementioned ensemble techniques over fuzzy clustering, an additional "hardening" process would be required for the fuzzy cluster assignments. This process may result in loss of information that is conveyed by the uncertainty measures of the relevant cluster assignments. This is particularly true for application settings where the underlying clustering algorithms access only a partial view of the data, such as in distributed data mining [27]. Yet, most of the existing cluster ensemble methods are based on crisp clusters. However, interesting departures from such work have recently been reported, including sCSPA, sMCLA and sHBGF (which are the fuzzy versions of the graph/hypergraph-based algorithms CSPA, MCLA and HBGF, respectively) [27].

Following such recent development, this paper presents a link-based hierarchical consensus-based approach for building ensembles of fuzzy c-means. The resulting algorithm is intended to be applied for clustering of big data. Different from ensembles of crisp clusters, the proposed approach allows direct handling of fuzzy clustering components, and generates a fuzzy partition as the final clustering outcome. In particular, the algorithm creates a fuzzy graph $< \{\widetilde{C}_1, ..., \widetilde{C}_n\}, \widetilde{L} >$ that $\widetilde{L}$ represents the fuzzy links between each pair of base fuzzy clusters $(\widetilde{C}_i, \widetilde{C}_j), i, j = 1, \cdots, n$, leading to a hierarchical structure of base clusters through the use of single-linkage agglomerative clustering. From this, the membership of each data point belonging to any final fuzzy cluster is calculated by simply aggregating its memberships to the respective base clusters.

In addition to the theoretical development of this fuzzy cluster ensemble approach, its potential application to clustering big data is also introduced in this paper. For this, an original multi-feature dataset is first divided into multiple one-dimensional subsets and individual clustering members are generated with respect to these subsets. Then, a set of fuzzy links between the resulting fuzzy base clusters are created, reflecting the similarities between fuzzy base clusters. Finally, the single-linkage agglomerative clustering method is implemented at the base cluster level (as opposite to the conventional use of this technique that is at the

data point level). Note that the computation of fuzzy c-means on one-dimensional datasets is much faster than that on multi-dimensional datasets whilst the clustering procedures on different subsets can be parallelised. Importantly, the number of base clusters is normally much less than the number of data points, which helps further reduce the overall computational complexity thanks to single-linkage clustering. Experimental results show that the hierarchical fuzzy cluster ensemble outperforms conventional clustering methods in terms of accuracy without sacrificing efficiency.

The rest of this paper is organised as follows. Section II introduces the basic concepts of the cluster ensemble framework upon which HFCE is based. Section III presents the HFCE approach in detail, including a discussion of its advantages over the existing cluster ensemble methods. Section IV describes the potential to exploit this method as a clustering tool for big data, through parallel implementation. Section V reports on the experimental set up and analyses the results. The paper is concluded in Section VI, with a discussion of interesting future research.

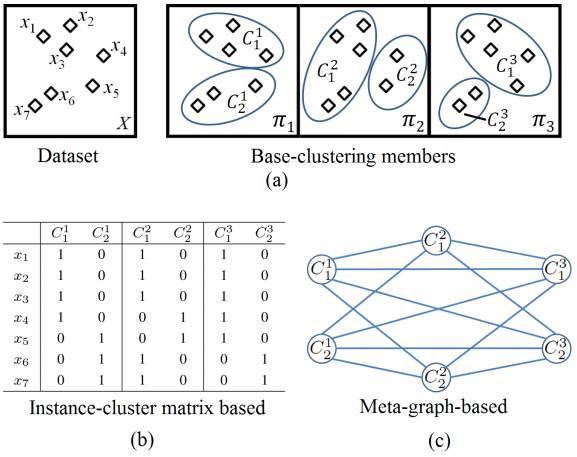
2. Cluster Ensemble and Fuzzy Cluster Ensemble

This section provides background context for the subsequent development, including the main concepts for conventional approach to crisp cluster ensembles and their fuzzy counterparts.

2.1. Cluster Ensemble

Formally, cluster ensemble can be described as follows. Let $X = \{x_1, \cdots, x_N\}$ be a set of N data points and $\Pi = \{\pi_1, \cdots, \pi_m, \cdots, \pi_M\}$ be M ensemble members. Applied to X, each ensemble member returns a set of clusters $\pi_m = \{C_1^m, \cdots, C_k^m, \cdots, C_{K_m}^m\}$ such that $\bigcup_{k=1}^{K_m} C_k^m = X$, where K_m is the number of clusters constructed by π_m. The clusters generated by all ensemble members together form a set of base clusters for the ensemble: $\{C_1, \cdots, C_n\} = \bigcup_{m=1}^{M} \pi_m$, where $n = \sum_{m=1}^{M} K_m$. The task of cluster ensemble is to find a new clustering result π^* given a data set X which summarises the information embedded in Π.

As indicated previously, two key procedures are involved in the development of a cluster ensemble. First, ensemble members are generated, typically by artificially diversifying methods for parameter settings and data re-sampling. Second, a consensus function is then applied on those ensemble members to generate the final clustering result. A consensus function can be generally viewed as a map from a set of ensemble members to one final partition of the original data $f : \Pi \rightarrow \pi$. A variety of consensus functions that are readily available may be applied to derive the required final data partition.



	C_1^1	C_2^1	C_1^2	C_2^2	C_1^3	C_2^3
x_1	1	0	1	0	1	0
x_2	1	0	1	0	1	0
x_3	1	0	1	0	1	0
x_4	1	0	0	1	1	0
x_5	0	1	0	1	1	0
x_6	0	1	1	0	0	1
x_7	0	1	1	0	0	1

Fig. 1. Examples of Cluster Ensemble Representation

Most of the consensus functions utilise an intermediate data structure which aggregates the information contained by ensemble members. Given an ensemble as shown in Fig. 1(a), two common types of such a data structure: the instance-cluster matrix-based and the meta-graph-based are illustrated in Fig. 1(b) and Fig. 1(c), respectively. Normally, a feature-based consensus function is directly applied to implement the former to achieve the final partition of the original data, with ensemble members working as a set of features [2,14]. The latter takes the view that an ensemble may be represented as a graph, where the nodes are base clusters or data points and links between them define the relationships holding amongst the clusters and/or points [9,31]. In this approach, additional computation is usually needed to calculate the extra information contained within the graph, such as the weights on the links between the base clusters.

In implementing the meta-graph-based approach, once a graph is generated, graph partition methods such as METIS [18] can be utilised to obtain the final data partition effectively. For example, the weight on the link between two base clusters C_i and C_j in a given meta-graph can be calculated as the binary Jac-

card similarity coefficient:

$$w(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}. \quad (1)$$

Another advantage of using meta-graph is that similarities (or indeed dissimilarities) between base clusters can be readily computed. Exploiting the similarity relations between base clusters, both within and across ensemble members, offers useful insight for producing quality final data partition [14].

2.2. Fuzzy Cluster Ensemble

If a crisp clustering algorithm such as k-means is used in the generation of base clusters, the association degree of a data point belonging to a specific cluster is either 1 or 0. However, there are other popular clustering algorithms such as EM and fuzzy c-means that naturally produce clusters of data with uncertain boundaries. Without loss of generality, fuzzy c-means is selected herein to generate the fuzzy base clusters due to its effectiveness in generating fuzzy partitions and also its availability. Each cluster in a fuzzy partition $\widetilde{\pi}$ is a fuzzy set $\widetilde{C}_k, k = 1, \cdots, K$ where $\widetilde{C}_k(x_t) \in [0, 1]$ represents the degree of a data point $x_t \in X$ belonging to the corresponding fuzzy cluster. Usually, this degree is normalised with all the clusters in a partition to satisfy that $\sum_{k=1}^{K} \widetilde{C}_k(x_t) = 1$.

Formally, a fuzzy (or soft) cluster ensemble can be described as follows [27]. Let $X = \{x_1, \cdots, x_N\}$ be a set of N data points and $\Pi = \{\widetilde{\pi}_1, \cdots, \widetilde{\pi}_m, \cdots, \widetilde{\pi}_M\}$ be M fuzzy ensemble members. Each ensemble member returns a set of fuzzy clusters $\widetilde{\pi}_m = \{\widetilde{C}_1^m, \cdots, \widetilde{C}_k^m, \cdots, \widetilde{C}_{K_m}^m\}$ such that $\sum_{k=1}^{K_m} \widetilde{C}_k^m(x_i) = 1$, where K_m is the number of fuzzy clusters constructed by that member. The fuzzy clusters generated by all ensemble members together form a set of fuzzy base clusters for the ensemble: $\{\widetilde{C}_1, \cdots, \widetilde{C}_n\} = \bigcup_{m=1}^{M} \widetilde{\pi}_m$, where $n = \sum_{m=1}^{M} K_m$. For each $x_t \in X$ and each ensemble member $\widetilde{\pi}_m \in \Pi$, $\widetilde{C}_k^m(x_t) \in [0, 1]$ denotes the degree of which the data point x_t belongs to the fuzzy cluster $\widetilde{C}_k^m$. An example of the so-called instance-cluster matrix of a fuzzy cluster ensemble is shown in Table. 1. The task of a fuzzy cluster ensemble is: for a given dataset X, find a new partition π^* which summarises the information embedded in the whole cluster ensemble Π. Such a cluster ensemble technique does not specify whether the final clusterings should be crisp or fuzzy. As to be shown later, given a dataset the proposed HFCE will produce a fuzzy partition $\widetilde{\pi}^*$ of it as the ensemble outcome.

Note that a key difference between crisp cluster ensemble and fuzzy cluster ensemble is that the latter works on fuzzy clusters. If the fuzzy base clusters are defuzzified into crisp clusters, many of the consensus functions designed for crisp cluster ensemble can be borrowed for use in building fuzzy cluster ensembles. However, valuable information may be lost in defuzzification and hence, the quality of the ensemble may be adversely affected [27]. For example, in crisp cluster ensemble, the instance-cluster matrix is sparse (see Fig. 1) and its contained information for a cluster ensemble is incomplete [14]. This problem is reflected in the meta-graph-based approach such that there will be no links between those base clusters generated by the same ensemble member (since there are absolutely no shared points between different crisp clusters). Thus, although fuzzy base clusters contain more information, conventional consensus functions for crisp cluster ensemble cannot directly make use of such information. It is due to this observation that fuzzy or soft cluster ensembles have been introduced in the literature [7,27,38] and followed on herein.

Table 1
Example of Non-binary Instance-cluster Matrix

	C_1^1	C_2^1	C_1^2	C_2^2	C_1^3	C_2^3
x_1	0.6	0.4	0.6	0.4	0.6	0.4
x_2	0.8	0.2	0.8	0.2	0.8	0.2
x_3	0.5	0.5	0.9	0.1	0.8	0.2
x_4	0.7	0.3	0.2	0.8	0.8	0.2
x_5	0.2	0.8	0.4	0.6	0.6	0.4
x_6	0.4	0.6	0.6	0.4	0.1	0.9
x_7	0.0	1.0	0.7	0.3	0.1	0.9

3. Hierarchical Fuzzy Cluster Ensemble

The proposed hierarchical cluster ensemble algorithm starts by creating ensemble members using fuzzy c-means on each feature in dataset. The resulting fuzzy base clusters and the links between them are represented in a fuzzy graph. The idea of hierarchical clustering is then employed to iteratively group the nodes based on the fuzzy links, in order to create the hierarchical structure that leads to the final clusters. It also yields instance-wise fuzzy cluster membership estimation, which may be defuzzified such that each data

point belongs to just one final cluster if required. The following details the key operations of this algorithm.

3.1. Feature-based Generation of Ensemble Members

In general, no constraints are necessarily imposed over the generation procedure through which the clustering partitions are obtained. In fact, different (component) clustering algorithms or the same algorithm with different parameter settings can be applied. If so desired, even different representations for data points, different subsets of data points or their projections on different subspaces may be used [8,40]. To perform a conventional cluster ensemble task, a moderate-sized dataset can be clustered several times in order to obtain ensemble members.

In dealing with big data, however, the computational overheads of running a single clustering procedure on a complete dataset may already be intolerable, multiple executions of clustering on the whole dataset are impractical. Reducing the complexity of dataset for each clustering member offers a reasonable way of solving this problem. In this work, an M-dimensional dataset is divided into M one-dimensional subsets, and M times of fuzzy c-means are carried out on those one-dimensional subsets.

In practice, projecting data onto different subspaces or choosing different subsets of features may lose information (e.g., correlations between features) which can be important to detect the underlying patterns of the data [16]. Unfortunately, this is also true in the proposed method where the qualities of the resulting individual ensemble members are generally not so good as those created from the direct use of all accessible features. Despite of this observation, as demonstrated in the existing ensemble leaning frameworks [29,36,42], relatively weak component results are still commonly used. Whilst individual ensemble members may be simple, if jointly utilised in conjunction with an appropriate consensus function, weak base clusters are capable of producing high quality ensemble results.

Such an individual feature-based partition strategy however, has further limitations in dealing with datasets that have redundant or interactive features. If the features are redundant, HFCE may produce redundant base clusters accordingly. Also, if certain individual features are interactive with each other, the useful information embedded in the interactions will be lost. A possible approach to solving these problems is to use feature selection or grouping techniques [43,44] in guiding the partition of the original data. However, the assessment of any redundancy and interaction amongst features incurs additional computational cost in the generation of base clusters and therefore, improvement over these issues can be rather time-consuming when dealing with datasets with a high dimensionality. Thus, in the current design and implementation of HFCE, advanced feature partition strategies are sacrificed to compensate for its executive speed. Finding a rapid feature partition algorithm to support HFCE remains active as further research.

It may be difficult to know a-priori which base clustering algorithm(s) will be appropriate for a given clustering problem. It is generally advisable and also, a common practice to employ those clustering algorithms that are known to be able to reflect and make use of most information embedded in the data. This is obvious as the more information each clustering member holds, the more information there is for the consensus function to work on. Based on this understanding, fuzzy c-means, which is able to retain the non-binary memberships of each data point to all clusters is adopted as the base algorithm for the generation of ensemble members in this work.

3.2. Similarity between Fuzzy Base Clusters

In the above proposed strategy for ensemble member generation the base clusters are created by partitioning the dataset with respect to different individual features. However, all data points used come from the same original dataset. As such, the resulting base clusters may share certain points. These shared data points naturally create linkages amongst base clusters and therefore, it is possible to estimate the similarity of any base cluster pair by exploring the underlying link information [32].

Note that the concept of a graph formulated from a set of base clusters and a set of weighted links between them has been introduced previously, as of [14]. Given a cluster ensemble as defined in Section 2.1, a graph $< V, L >$ can be constructed where $V = \bigcup_{m=1}^{M} \pi_m = \{C_1, \cdots, C_n\}, n = \sum_{m=1}^{M} K_m$ is the set of vertices each representing a base cluster, and L is a set of weighted links between the clusters. In particular, the weight on the link between two base clusters C_i and $C_j (i, j = 1, \cdots, n)$, can be defined as given in Eqn (1). In order to retain more information from fuzzy clustering components and reflect the interactions between different features which are embedded in the original dataset, a fuzzy graph of fuzzy c-means ensemble is employed here.

Formally, given a set of fuzzy base clusters $V = \{\widetilde{C}_1, \cdots, \widetilde{C}_n\}$ on a dataset $\{x_1, \cdots, x_N\}$, a fuzzy graph $< V, \widetilde{L} >$ is defined on V with $\widetilde{L}$ being a fuzzy set of links defined on $V \times V$. The membership of a link $(\widetilde{C}_i, \widetilde{C}_j), i, j = 1, \cdots, n$ to the fuzzy set that represents the fuzzy relation $\widetilde{L}$ is computed by

$$\mu_{\widetilde{L}}(\widetilde{C}_i, \widetilde{C}_j) = \frac{\sum_{t=1}^{N} \min(\widetilde{C}_i(x_t), \widetilde{C}_j(x_t))}{\sum_{t=1}^{N} \max(\widetilde{C}_i(x_t), \widetilde{C}_j(x_t))} \quad (2)$$

where $\widetilde{C}_i(x_t)$ indicates the degree of a data point x_t belonging to a fuzzy base cluster $\widetilde{C}_i$. If $\max_{t=1}^{N}(\widetilde{C}_i(x_t), \widetilde{C}_j(x_t)) = 0$, then $\mu_{\widetilde{L}}(\widetilde{C}_i, \widetilde{C}_j) = 1$. The resultant fuzzy graph can also be represented as a meta-graph. Similar to Fig. 1(c), each fuzzy base-cluster can be represented as a node and the membership of a link to $\widetilde{L}$ can be represented as the strength or weight of that link. Obviously, $\mu_{\widetilde{L}}(\widetilde{C}_i, \widetilde{C}_j) \in [0, 1]$, $\mu_{\widetilde{L}}(\widetilde{C}_i, \widetilde{C}_i) = 1$ and $\mu_{\widetilde{L}}(\widetilde{C}_i, \widetilde{C}_j) = \mu_{\widetilde{L}}(\widetilde{C}_j, \widetilde{C}_i)$. The degree assigned to the link connecting fuzzy clusters $\widetilde{C}_i$ and $\widetilde{C}_j$ is thus defined in accordance with the proportion of their overlapping degree on all data points in X. In so doing, for two fuzzy base clusters within the same ensemble member, the weight on the link between them is possible to be of a non-zero value. As such, in general, each fuzzy base cluster may have a link to all the other fuzzy base clusters.

By definition of the link weights, the fuzzy degree of any given link intuitively captures the underlying similarity between the corresponding two fuzzy base clusters. These fuzzy links are of particular significance in this work. Since the ensemble members are generated from one-dimensional subsets, information on the interactions or correlations between features is lost in compromise with the gain of computing time. However, by employing a link-based consensus function that makes use of similarities between base clusters, such information can be (re-)captured.

The similarities between base clusters carry the information of how close they are to one another, and this information is useful to merge redundant base clusters. In crisp cluster ensembles, base clusters within the same base clustering member do not have common data points with each other, that is, $\forall C_k^m, C_l^m \in \pi^m$, if $k \neq l$ then $C_k^m \cap C_l^m = \emptyset$. The weights on those links between the clusters within the same base clustering member are of a value of zero. In this case, further refinement will have to be carried out in an effort to estimate the similarities between base clusters within the same ensemble member. For example, connected-triple links cross ensemble members are computed and then exploited for this purpose in the work of [14]. Fortunately, in fuzzy c-means ensemble, non-zero weighted links exist not only between those base clusters within a single base clustering member, e.g., $\exists \mu_{\widetilde{L}}(\widetilde{C}_k^m, \widetilde{C}_l^m) > 0$, but also between base clusters cross different base-clustering members, e.g., $\exists \mu_{\widetilde{L}}(\widetilde{C}_k^m, \widetilde{C}_l^n) > 0, m \neq n$. Since no additional refinement is needed (as otherwise needed for the crisp case), the similarity measures can be readily computed, making significant savings in time and memory space.

3.3. Base Cluster Grouping via Hierarchical Clustering

In this step, fuzzy base clusters are grouped into a certain number of final clusters to become the output of the ensemble. In order to group the fuzzy base clusters they are artificially treated as data instances and those original data points given in the dataset that belong to a base cluster are regarded as a feature for the artificial data instance. In other words, the "instance-cluster matrix" in Table 1 is transposed to a "cluster-instance matrix", which generally speaking, has a large number of features but a relatively smaller number of instances involved than the real instance-cluster matrix.

In the existing work on fuzzy cluster ensembles, each base cluster maintains the non-binary membership values of all those data points belonging to it. This makes fuzzy base clusters more informative but more storage-consuming than their crisp counterparts. Fortunately, link-based clustering approaches such as single-linkage clustering [30] do not need to re-access the original memberships of the data points once the similarity matrix is obtained, making them less sensitive to high-dimensional data. This is important in an effort to deal with the grouping of base clusters, since otherwise iteratively visiting fuzzy base clusters directly can be very time-consuming, if not prohibitive.

By using single-linkage hierarchical clustering algorithm, grouping fuzzy base clusters can be achieved without the need of updating the cluster centroids. A matrix $L = [l(i, j)]_{n \times n}$ can be constructed with the indices of its rows and columns representing the indices of base clusters, and each entry $l(i, j), i, j = 1, \cdots, n$, of the matrix representing the similarity value of the corresponding base clusters, i.e., the membership value of the fuzzy link between $\widetilde{C}_i$ and $\widetilde{C}_j$ as defined in Eqn. (2). From this, the grouping of the

base clusters depends upon L only, rather than upon the memberships of the original data points belonging to these base clusters. The subsequent steps of single-linkage clustering are quite simple after the similarities amongst base clusters are known: Applying a simple sorting procedure over those similarity values and then using a threshold or a given number of total clusters required to merge the clusters. Therefore, both the time and memory resources required for iteratively visiting instance-cluster matrix or the pair-wised similarity matrix are saved. Note that single-linkage clustering obtains exactly the same results by agglomerating small clusters into larger ones (bottom up) as by dividing larger clusters into smaller ones (top down) [11]. In the following, this method is referred to as single-linkage agglomerative clustering (SLAC).

Apart from SLAC, a multilevel scheme for partitioning irregular graphs, METIS [18] has been used to group base clusters in sMCLA. However, METIS can only produce a given number of balanced groups of clusters. Since each group is required to contain the same number of base clusters, this method is not suitable for use in the present approach where one-dimensional individual feature-based partition of the dataset is assumed. For example, in the situation where the dataset contains certain outlier values in one feature, a base cluster consists of only the outliers can be generated. Such a cluster should be grouped with itself rather than with any other normal ones. Therefore, algorithms which return absolutely balanced groups of clusters may damage the overall quality of the resultant cluster groups. In light of this observation and considering both the time complexity and the quality of clustering, SLAC is selected to group base clusters in HFCE.

Recall the basic idea of hierarchical clustering, that is to build a tree of data clusters that are successively merged into similar groups, with each level of the resulting tree being a segmentation of the original data [17]. By applying SLAC to group fuzzy base clusters rather than data points or crisp clusters, each level of the resulting tree comprises groups of base fuzzy clusters. In order to obtain the overall data partition by the ensemble, an additional step which transforms the groups of fuzzy base clusters into the final fuzzy clusters of the original data points is needed.

3.4. Final Assignment of Data Points

At each level of the resultant hierarchical tree, all those base clusters contained within a certain cluster-group are collapsed to form one single fuzzy cluster. At the leaf level of the tree each fuzzy base cluster contains a membership value for every data point that is deemed to belong to the cluster. Such a membership for a given final fuzzy cluster is computed as the normalised mean of its memberships to all those base clusters that are grouped together. To produce a crisp final partition of the original data, each point is assigned to the cluster group to which it has the highest membership. Note that if so desired, other aggregation operators [33] rather than the average may also be employed to implement this of course.

Summarising the above development, the proposed hierarchical fuzzy cluster ensemble (HFCE) learning algorithm is given in Algorithm 1.

Algorithm 1 Hierarchical Fuzzy Cluster Ensemble

Inputs: $X = \{x_1, \cdots, x_t, \cdots, x_N\}, x_t = (a_1^t, \cdots, a_M^t) \in \Re^M$: a dataset of N instances and M features;
$K_1, \cdots, K_m, \cdots, K_M$: number of base clusters in each ensemble member;
K : final number of clusters.
Outputs: $\widetilde{\pi}^* = \{\widetilde{C}_1^*, \cdots, \widetilde{C}_K^*\}$: a fuzzy partition of X.

1: **for** $m = 1 : M$ **do**
2: create sub-dataset $X_m = \{a_m^1, \cdots, a_m^N\}$
3: create ensemble member $\widetilde{\pi}_m = \{\widetilde{C}_1^m, \cdots, \widetilde{C}_{K_m}^m\}$ using fuzzy c-means on X_m
4: **end for**
5: merge the ensemble members to create a set of fuzzy base clusters $V = \{\widetilde{C}_1, \cdots, \widetilde{C}_i, \cdots, \widetilde{C}_n\} = \bigcup_{m=1}^M \pi_m$, where $n = \sum_{m=1}^M K_m$
6: **for** $i = 1 : n-1$ **do**
7: **for** $j = i+1 : n$ **do**
8: $\mu_{\widetilde{L}}(\widetilde{C}_i, \widetilde{C}_j) = \frac{\sum_{t=1}^N \min(\widetilde{C}_i(x_t), \widetilde{C}_j(x_t))}{\sum_{t=1}^N \max(\widetilde{C}_i(x_t), \widetilde{C}_j(x_t))}$
9: **end for**
10: **end for**
11: create a partition $\pi^V = \{C_1^V, \cdots, C_k^V, \cdots C_K^V\}$ on V based on $\widetilde{L}$ using hierarchical clustering
12: **for** $k = 1 : K$ **do**
13: $\mu'_{\widetilde{C}_k^*}(x_t) =$ average of $\{\widetilde{C}_i(x_t) | \widetilde{C}_i \in C_k^V\}$
14: **end for**
15: normalise $\mu'_{\widetilde{C}_k^*}(x_i)$ to $\mu_{\widetilde{C}_k^*}(x_t)$, such that $\sum_{k=1}^K \mu_{\widetilde{C}_k^*}(x_t) = 1$

4. Application to Big Data

Although much effort has been made in the development of cluster ensembles, the application of cluster ensemble techniques is still at an early stage [13]. Little has been successfully done for big data. This section proposes an initial idea as to how HFCE may be potentially utilised to handle big data clustering, based on an investigation into its time complexity.

For a dataset with N points and M features, the time complexity of the original fuzzy c-means is $O(MNK_m)$ where K_m is the number of clusters [20]. The hierarchical agglomerative clustering has a time complexity of $O(N^2 logN)$ [15]. Since the agglomerative clustering is employed only for grouping base clusters in HFCE, its use leads to a complexity of $O(n^2 logn)$, where $n = \sum_{m=1}^{M} K_m$ is the total number of fuzzy base clusters generated. To calculate the final fuzzy partition the algorithm also involves an additional time complexity of $O(nNK)$.

For big data it may be difficult to expect that Algorithm 1 can be implemented on a single computer of moderate computational power. However, HFCE can be implemented in a parallel way. Suppose that there are $P+1$ computers within a certain parallel computer network. One computer acts as the host node, which is in charge of assigning tasks and collecting results [41], and the other P computers work as the real computing nodes. Then, such a parallel computation network can be applied to implement the three main components of HFCE, reducing its overall time complexity. The parallelisation is outlined below:

1) Generation of ensemble members: Each ensemble member is denoted by π_m, where $m = 1, \cdots, M$ and each π_m corresponds to a certain feature of the dataset. If $P < M$, then the host node first assigns $\pi_1, \cdots, \pi_P$ to the P computing nodes. Once any of the P ensemble member is generated, the host node orderly assigns the next component of $\pi_{P+1}, \pi_{P+2}, \cdots$ to the free computing nodes. This process iterates until π_M is reached. If $P \geq M$, the assignment is straightforward. Since fuzzy c-means only takes few iterations to converge on one-dimensional data, the burden of each computing nodes is very low.

2) Single-linkage-based grouping of fuzzy base clusters: A number of parallel versions of SLAC have been proposed in the literature (e.g., [26,28]). For simplicity, an intuitive parallel method (not the best time-saving one) is introduced here. As any pair of fuzzy base-clusters have in general, a fuzzy link between them, the membership values of all the links $\mu_{\widetilde{L}}(\widetilde{C}_i, \widetilde{C}_j)$ can be collectively represented as a pairwise similarity matrix $L_{n\times n}$. Since this matrix is symmetric and the similarity of one element to itself is not considered in SLAC, $\frac{n(n-1)}{2}$ similarity values are needed to be computed. The host node decomposes the task of calculating these similarity values into the P computing nodes. Note that only a limited number of fuzzy base clusters are generated for each feature, and that this number is generally much smaller than the number of data points. Thus, after $L_{n\times n}$ is obtained, the parallelisation of following steps in SLAC for base cluster grouping may not be necessary, but optional for a powerful computing node.

3) Computation of the memberships of data points to the final clusters: Similar to the working of step 1), the final clustering result can be represented as a matrix $A_{N\times K}$ (as illustrated in Table 1). Given that $N > P, p = \lfloor N/P \rfloor$, the host node can decompose $A_{N\times K}$ into P disjointed sub-matrices $A_{[1,\cdots,p]\times K}$, $A_{[p+1,\cdots,2p]\times K}$, $\cdots$, $A_{[p\times(P-1),\cdots,N]\times K}$. The required membership calculations regarding the resulting P sub-matrices can then be assigned to the P computing nodes, respectively.

5. Experimentation and Evaluation

This section presents an experimental evaluation of the proposed work. It first outlines the set-up of the experiments carried out and then discusses the results obtained. One experimentation is designed to test the quality of clusters which are generated using HFCE, in comparison to those produced by the use of alternative approaches, and another to show the time efficiency of HFCE in contrast with the original fuzzy c-means and single-linkage agglomerative clustering.

5.1. Experimental Set-up

To evaluate the performance of the proposed approach, the algorithm is tested over nine datasets obtained from the UCI benchmark repository [3], where the underlying true labels of the data points are known (which are not explicitly used in the cluster ensemble learning process but in the computation of clustering accuracy). The details of these datasets are summarised in Table 2.

In HFCE, fuzzy c-means is used to implement fuzzy ensemble members. In each ensemble member, the number of base clusters $K_m, m = 1, 2, \cdots, M$ is set to the number of given classes of the dataset. The clus-

Table 2
Summary of Datasets Used

Datasets	# Instances	# Attributes	# Classes
Iris	150	4	3
Sonar	208	60	2
Statlog Heart	270	13	2
Parkinsons	195	22	2
Ionosphere	351	34	2
Pima Indians Diabetes	768	8	2
Yeast	1484	8	10
Statlog Landsat Satellite	4435	36	6
Spambase	4601	57	2

ter centroids are randomly initialised in each run. The single-linkage clustering technique is selected to implement base cluster grouping. It organises base clusters into a hierarchical tree using the underlying similarity matrix given in Eqn. 2. For comparison, an ensemble of crisp clusters (HCCE) with a similar underlying mechanism to that of HFCE is also implemented. To have a common ground for this comparative study, the base clusters used in HCCE are those defuzzified from the base fuzzy c-means used in HFCE and the number of final clusters on each dataset is again set to that of its true classes. The output of HFCE is defuzzified by assigning a data point to the cluster to which it reaches the maximum membership.

5.2. Clustering Quality

In order to gauge clustering quality, two types of criterion are usually employed, measuring how well a clustering partitions the given data into the underlying groupings, namely, the internal and external criteria [13]. In particular, the goodness of a clustering ensemble is estimated using the averaged Silhoutte index [22] which measures the compactness of resultant clusters without referring to the ground truth (internal). If however, the class labels are available for all the data involved in the experiments, the final clustering results can be evaluated using the accuracy which measures how well the clusters match the given true labels of the data points (external). In this experiment, the quality of the final clustering outcomes is also assessed using these two criteria.

The resultant averaged Silhoutte index and clustering accuracy rates are shown in Tables 3 and 4 respectively, where the best-two results on each dataset are highlighted in boldface and each number in the table is an averaged value based on 50 runs. To validate the significance of the experimental results, paired-t tests are carried out. The baseline for comparison is the result of running HFCE. In this table, the sign "(-)" indicates that the corresponding result of HCCE is significantly worse ($p < 0.05$), while "(*)" indicates that the corresponding one is significantly better. In order to compare the ensemble-based clustering methods with the conventional clustering methods, the results of fuzzy c-means (FCMC) and single-linkage agglomerative clustering (SLAC) are also included.

Experimental results show that HFCE achieves better compactness and accuracy than HCCE on most of the datasets. This indicates that the information embedded in fuzzy base clusters are more effective to generate final ensemble partitions then that embedded in crisp base clusters. Also, the final clusters generated by HFCE lead to better accuracies than SLAC over seven datasets. However, the performance of HFCE is not significantly better than FCMC in general. A likely reason is that the tested datasets are sensitive to the interaction of features, while the similarities amongst fuzzy base clusters cannot completely capture the interaction of features in those datasets. Importantly, HFCE only employs ensemble members which generate base clusters each involving just one feature.

5.3. Time Efficiency

This set of experiments is to empirically check the time efficiency of HFCE, as compared to that of the original SLAC and FCMC. To ensure that the results are easy to analyse, the simple iris dataset is used. However, to vary the scale of each experiment, the original dataset is artificially enlarged either horizontally (by duplicating features), or vertically instances (by duplicating data points), or both. In the following presentation, the original dataset is denoted as $[D]$, its horizontally double-sized dataset $[D\ D]$ is denoted as $[D]^{\times 2}$, and its vertically double-sized dataset $\begin{bmatrix} D \\ D \end{bmatrix}$ is denoted as $[D]_{\times 2}$, etc. The experiments are carried out on a computer with Inter(R) Core(TM)2 Duo 3.00 GHz $\times$ 2 CPU, 4 GB RAM, and Windows 7 (64-bit) operation system. All three methods under comparison are implemented in series with Matlab 7.11 win64 version. Each point in these figures is an averaged value of 50 runs.

Figures 2, 3 and 4 present the time cost of running each of the aforementioned three methods (HFCE, SLAC and FCMC) in response to the increase of the number of features, the number of data points and the

Table 3
Comparison of Averaged Silhoutte Index ($[-1, 1]$)

Dataset	HFCE	HCCE	SLAC	FCMC
Iris	0.678±0.003	0.267±0.243(-)	**0.700±0.000**(*)	**0.685±0.000**(*)
Sonar	0.136±0.020	0.175±0.093(*)	**0.446±0.000**(*)	**0.267±0.009**(*)
Statlog Heart	0.184±0.000	0.204±0.093	**0.261±0.000**(*)	**0.342±0.000**(*)
Parkinsons	0.349±0.000	0.354±0.223	**0.792±0.000**(*)	**0.417±0.000**(*)
Ionosphere	0.230±0.000	0.219±0.086	**0.496±0.000**(*)	**0.331±0.000**(*)
Pima Indians Diabetes	**0.469±0.000**	0.283±0.081(-)	**0.648±0.000**(*)	0.381±0.012(-)
Yeast	**0.032±0.036**	-0.089±0.042(-)	**0.640±0.000**(*)	-0.048±0.034(-)
Statlog Landsat Satellite	**0.289±0.054**	-0.154±0.067(-)	-0.015±0.000(-)	**0.471±0.000**(*)
Spambase	**0.644±0.000**	0.052±0.028(-)	**0.875±0.000**(*)	0.140±0.007(-)

Table 4
Comparison of Accuracy (%)

Dataset	HFCE	HCCE	SLAC	FCMC
Iris	**89.89±0.25**	65.24±9.48(-)	66.67±0.00(-)	**89.33±0.00**(-)
Sonar	54.33±0.00	**59.54±6.05**(*)	53.37±0.00(-)	**55.22±0.48**(*)
Statlog Heart	**67.41±0.00**	60.93±6.33(-)	55.93±0.00(-)	**79.26±0.00**(*)
Parkinsons	75.38±0.00	75.38±0.00	75.38±0.00	75.38±0.00
Ionosphere	**62.61±0.00**	55.48±3.14(-)	50.87±0.00(-)	**74.78±0.00**(*)
Pima Indians Diabetes	65.10±0.00	**68.64±3.21**(*)	65.23±0.00(*)	**66.67±0.00**(*)
Yeast	**41.96±3.59**	33.66±1.28(-)	32.35±0.00(-)	**43.01±0.13**
Statlog Landsat Satellite	**61.98±0.76**	40.35±3.95(-)	24.28±0.00(-)	**72.51±0.00**(*)
Spambase	**60.75±0.00**	60.60±0.00(-)	60.62±0.00(-)	**76.71±0.00**(*)

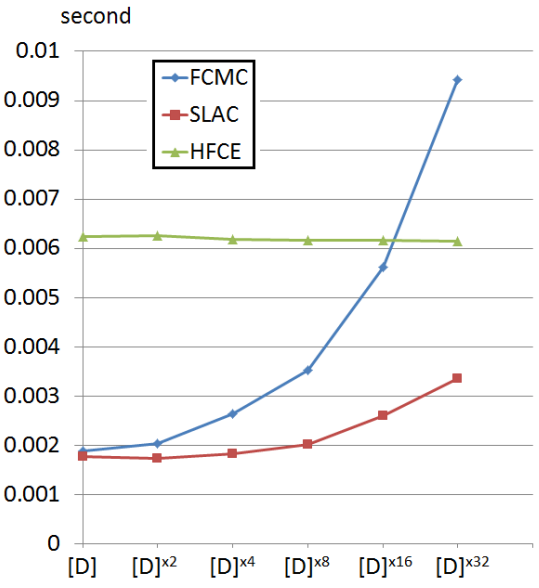


Fig. 2. Time Cost vs. Increase of Features

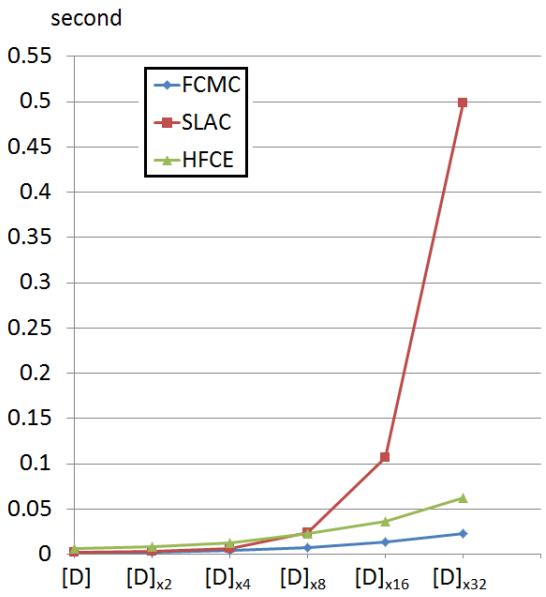


Fig. 3. Time Cost vs. Increase of Instances

number of both factors, respectively. It is clear that the execution time of these methods generally increases along with the increase of data size. However, HFCE shows a more stable performance than its counterparts when both the number of features and instances are increased (see Fig. 4). This shows that the use of feature-

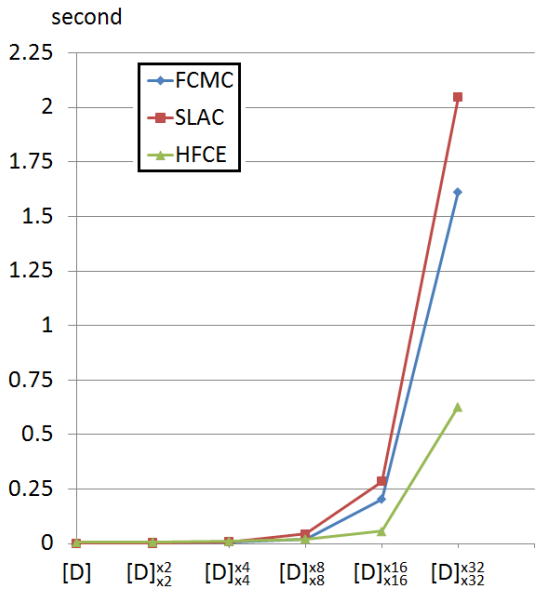


Fig. 4. Time Cost vs. Increase of Both Features and Instances

based partition of dataset and the pairwise similarity matrices entails more time efficiency in the proposed hierarchical fuzzy c-means ensemble, which in turn indicates the potential of HFCE in dealing with big data.

Note that the outcome of using HFCE on datasets with increased features seems to be more stable as compared with FCMC (Fig. 2). An intuitive explanation is that HFCE only computes the one-dimension distance between data points, which makes it far less sensitive to the "curse of dimensionality". However, HFCE still suffers from the increase of data points to a certain extent. Nevertheless, it is not so drastic as the algorithms which have a time complexity of $O(N^2)$ or above, as reflected in Fig. 3. Thus, the proposed HFCE takes the advantage of SLAC when dealing with an increasing number of features and that of fuzzy c-means when dealing with an increasing number of data points. The downside is that although reasonable, the accuracy of HFCE is not so high as that achievable by the original fuzzy c-means in general. Nevertheless, HFCE allows for higher time efficiency than fuzzy c-means without a drastic loss of clustering quality.

6. Conclusion

This paper has presented an approach for feature-based ensemble member generation and for link-based hierarchical base clusters grouping, in building hierarchical fuzzy (c-means) cluster ensembles. The proposed work takes the advantage of fuzzy c-means in that each data point can have a membership to all clusters. It also takes the advantage of hierarchical clustering in that the iterative access of data points is replaced by the computation of pair-wised similarity measures. Experimental results on nine popular benchmark datasets indicate that the proposed approach generally outperforms its crisp counterparts (HCCE and single-linkage agglomerative clustering). Furthermore, it also has the potential to process big data as the approach entails a higher time efficiency compared to the original fuzzy c-means and hierarchical clustering.

Whilst promising, the present work opens up an avenue for further investigation. For instance, many other ensemble member generating methods such as resampling may also be applied for clustering. It would be useful to examine the performance of the proposed fuzzy graph using different similarity measures and also, the effects of the suggested parallel implementation. Finally, it will be very interesting to implement this work in a real problem setting which involves big data.

7. Acknowledgements

The authors would like to thank the reviewers and the editor for providing useful comments that have helped revise this work. The first author is grateful to Aberystwyth University for providing a full-fees PhD scholarship in support of his research.

References

[1] Charu C Aggarwal and Chandan K Reddy. *Data Clustering: Algorithms and Applications*. CRC Press, 2013.

[2] Hanan G Ayad and Mohamed S Kamel. On voting-based consensus of cluster ensembles. *Pattern Recognition*, 43(5):1943–1953, 2010.

[3] K Bache and M Lichman. UCI machine learning repository, 2013.

[4] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984.

[5] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[6] Ren Diao, Fei Chao, Taoxin Peng, N. Snooke, and Qiang Shen. Feature selection inspired classifier ensemble reduction. *Cybernetics, IEEE Transactions on*, 44(8):1259–1268, 2014.

[7] Carlotta Domeniconi and Muna Al-Razgan. Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(4):17, 2009.

[8] Xiaoli Z Fern and Carla E Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning*, pages 186–193, 2003.

[9] Xiaoli Z Fern and Carla E Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the Twenty-first International Conference on Machine Learning*, page 36, 2004.

[10] Ana LN Fred and Anil K Jain. Combining multiple clusterings using evidence accumulation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):835–850, 2005.

[11] John C Gower and GJS Ross. Minimum spanning trees and single linkage cluster analysis. *Applied Statistics*, pages 54–64, 1969.

[12] Stefan T Hadjitodorov, Ludmila I Kuncheva, and Ludmila P Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006.

[13] Natthakan Iam-On and Tossapon Boongoen. Comparative study of matrix refinement approaches for ensemble clustering. *Machine Learning*, pages 1–32, 2013.

[14] Natthakan Iam-On, Tossapon Boongoen, Simon Garrett, and Chris Price. A link-based approach to the cluster ensemble problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2396–2409, 2011.

[15] A K Jain, M N Murty, and P J Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 1999.

[16] Shuyuan Jin, Daniel So Yeung, and Xizhao Wang. Network intrusion detection in covariance feature space. *Pattern Recognition*, 40(8):2185–2197, 2007.

[17] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[18] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar. Multilevel hypergraph partitioning: applications in vlsi domain. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 7(1):69–79, 1999.

[19] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[20] John F Kolen and Tim Hutcheson. Reducing the time complexity of the fuzzy c-means algorithm. *Fuzzy Systems, IEEE Transactions on*, 10(2):263–267, 2002.

[21] Ludmila I Kuncheva and Stefan Todorov Hadjitodorov. Using diversity in cluster ensembles. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 2, pages 1214–1219. IEEE, 2004.

[22] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 911–916. IEEE, 2010.

[23] Piyush Malik. Governing big data: principles and practices. *IBM Journal of Research and Development*, 57(3/4):1–1, 2013.

[24] Raymond T. Ng and Jiawei Han. Clarans: A method for clustering objects for spatial data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 14(5):1003–1016, 2002.

[25] Nam Nguyen and Rich Caruana. Consensus clusterings. In *Data Mining, 2007. Seventh IEEE International Conference on*, pages 607–612. IEEE, 2007.

[26] Clark F Olson. Parallel algorithms for hierarchical clustering. *Parallel Computing*, 21(8):1313–1325, 1995.

[27] Kunal Punera and Joydeep Ghosh. Consensus-based ensembles of soft clusterings. *Applied Artificial Intelligence*, 22(7-8):780–810, 2008.

[28] Edie M Rasmussen and Petter Willett. Efficiency of hierarchic agglomerative clustering using the icl distributed array processor. *Journal of Documentation*, 45(1):1–24, 1989.

[29] Qiang Shen, Ren Diao, and Pan Su. Feature selection ensemble. In Andrei Voronkov, editor, *Alan Turing Centenary*, pages 289–306, 2012.

[30] Peter HA Sneath. The application of computers to taxonomy. *Journal of General Microbiology*, 17(1):201–226, 1957.

[31] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[32] Pan Su, Changjing Shang, and Qiang Shen. Link-based approach for bibliometric journal ranking. *Soft Computing*, 17(12):2399–2410, 2013.

[33] Pan Su, Changjing Shang, and Qiang Shen. Owa aggregation of fuzzy similarity relations for journal ranking. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–7, 2013.

[34] Alexander Topchy, Anil K Jain, and William Punch. Clustering ensembles: Models of consensus and weak partitions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1866–1881, 2005.

[35] Ran Wang, Yu-Lin He, Chi-Yin Chow, Fang-Fang Ou, and Jian Zhang. Learning elm-tree from big data based on uncertainty reduction. *Fuzzy Sets and Systems*, pp, 2014.

[36] Xizhao Wang, Ran Wang, Hui-Min Feng, and Hua-Chao Wang. A new approach to classifier fusion based on upper integral. *Cybernetics, IEEE Transactions on*, 44(5):620–635, May 2014.

[37] Xizhao Wang, Yadong Wang, and Lijuan Wang. Improving fuzzy c-means clustering based on feature-weight learning. *Pattern Recognition Letters*, 25(10):1123–1132, 2004.

[38] Linyun Yang, Hairong Lv, and Wenyuan Wang. Soft cluster ensemble based on fuzzy similarity measure. In *Computational Engineering in Systems Applications, IMACS Multiconference on*, pages 1994–1997, Oct 2006.

[39] Daniel S. Yeung and Xizhao Wang. Improving performance of similarity-based clustering by feature weight learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):556–561, 2002.

[40] Zhiwen Yu, Hau-San Wong, and Hongqiang Wang. Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics*, 23(21):2888–2896, 2007.

[41] Mohammed J Zaki. Parallel and distributed association mining: A survey. *IEEE Concurrency*, 7(4):14–25, 1999.

[42] Junhai Zhai, Hongyu Xu, and Yan Li. Fusion of extreme learning machine with fuzzy integral. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 21(supp02):23–34, 2013.

[43] Junhai Zhai, Mengyao Zhai, and Chenyan Bai. An improved algorithm for calculating fuzzy attribute reducts. *Journal of Intelligent and Fuzzy Systems*, 25(2):303–313, 2013.

[44] Junhai Zhai, Mengyao Zhai, and Xiaomeng Kang. Condensed fuzzy nearest neighbor methods based on fuzzy rough set technique. *Intelligent Data Analysis*, 18(3):429–447, 2014.

[45] J. Zhou, C. Chen, L. Chen, and H. Li. A collaborative fuzzy clustering algorithm in distributed network environments. *Fuzzy Systems, IEEE Transactions on*, PP(99):1–1, 2013.